УДК 004.4 http://doi.org/10.15407/pp2024.02-03.296

С.В. Поперешняк, Р.О. Скорик, Д.В. Купцов, Р.В. Кравченко

СИСТЕМА РОЗПІЗНАВАННЯ ОБЛИЧЧЯ ЛЮДИНИ У ВІДЕОПОТОЦІ

В роботі було проведено аналіз методів виявлення обличчя у відеопотоці та їхньої ефективності в реальному часі. Було з'ясовано, що сучасні алгоритми та попередньо натреновані моделі дозволяють з високою точністю розпізнавати обличчя, однак значним їхнім недоліком є, зокрема, вразливість до атак із використанням фальшивих облич. Тому було проведено також аналіз підходів до виявлення живих облич і можливості їхньої імплементації в системі. За допомогою об'єктно-орієнтованого підходу було спроєктовано та розроблено засіб для захоплення обличчя, отримання відеопотоку з різних джерел, виявлення невідомих та раніше захоплених облич у відеопотоці, розпізнавання живих облич. Система була адаптована для роботи в реальному часі із використанням GPU. В роботі було вдосконалено архітектуру конволюційної нейронної мережі для розпізнавання живих облич із створенням датасету із комбінації власних футажів та відкритих датасетів. Також розроблено користувацький інтерфейс для системи розпізнавання облич. У роботі покращено процедури ідентифікації та спрощення виявлення осіб на відео для працівників відділу безпеки підприємств шляхом впровадження методів розпізнавання із живих облич (liveness detection). Результатом дослідження було спроєктовано систему, призначену для виявлення, розпізнавання та виявлення живих облич у відеопотоці. Проаналізувавши відомі успішні програмні продукти, було виділено ніші, які потребують представлення нового рішення. На їх основі було розроблено функціональні та нефункціональні вимоги. Процес розпізнавання облич у відеопотоці було модифіковано шляхом впровадження власної моделі Liveness Detection.

Ключові слова: виявлення облич, розпізнавання облич, виявлення живих облич, глибоке навчання, відеопотік, Internet of Things, держкордон, потокова інформація, програмні комплекси, спостережуваність та керованість.

S. Popereshnyak, R. Skoryk, D. Kuptsov, R. Kravchenko

HUMAN FACE RECOGNITION SYSTEM IN VIDEO STREAM

In the work, an analysis of detection methods and faces in the video stream and their effectiveness in real time was carried out. Modern algorithms and pre-trained models have been found to be able to recognize faces with high accuracy, but their significant drawback is, in particular, vulnerability to attacks using fake faces. Therefore, the work also analyzed approaches to detecting living faces and the possibility of their implementation in the system. Using an object-oriented approach, a tool for face capture, receiving a video stream from various sources, detecting unknown and previously captured faces in a video stream, and recognizing live faces was designed and developed. The system has been adapted to work in real time using the GPU. The work improved the architecture of a convolutional neural network for recognizing living faces with the creation of a dataset from a combination of own footage and open datasets. Also, a user interface for the face recognition system was developed. The work improved identification procedures and simplified detection of persons on video for employees of the security department of enterprises by implementing liveness detection face recognition methods. As a result of the research, a system was designed, which is intended for detection, recognition and detection of living faces in a video stream. After analyzing the known successful software products, niches that need a new solution were identified. Based on them, functional and non-functional requirements were developed. The process of recognizing faces in the video stream has been modified by implementing our own Liveness Detection model.

Key words: face detection, face recognition, live face detection, deep learning, video streaming, Internet of Things, state border, streaming information, software complexes, observability and controllability.

Вступ

Система розпізнавання облич належить до біометричних систем безпеки, як і розпізнавання голосу, розпізнавання відбитків пальців та розпізнавання сітківки. Біологічний процес розпізнавання облич у людей є дуже цікавим, бо людина все ще може впізнати когось, навіть якщо він виріс або зазнав значних змін зовнішності. За допо-

могою комп'ютерних технологій це також можливо за умови достатньо якісного алгоритму. Проблема, яка часто виникає в системах розпізнавання облич, полягає в тому, що цю систему легко обдурити за допомогою підроблених облич або зображень людських облич, таких як фотографії, відео, маски та статуї, що називається спуфінг атакою. Це можливо, оскільки основним принципом розпізнавання облич є запам'ятовування унікальної інформації про обличчя, її кодування і порівняння з раніше закодованими результатами. Використання зображення людського обличчя для ідентифікації неможливо виявити без LD: зловмисники можуть використовувати фотографію людини, яка вже зареєстрована в системі, щоб обійти систему розпізнавання облич і отримати особисту вигоду. Для безпеки системи розпізнавання облич, нам потрібен алгоритм, який може визначити, чи є користувач справжнім чи підробленим, що і називається Liveness detection. На відміну від звичайного розпізнавання облич без цієї функції, яке просто бере дані з камери, не перевіряючи, чи це справжня людина, тут вирішений потенційний недолік: звичайне розпізнавання обличчя сприймає все, що перед ним, як справжнє людське обличчя, навіть якщо це просто зображення або об'єкт. Саме тут і з'являється функція виявлення живої людини, щоб виправити це.[1]

LD використовується в секторах, де висока безпека є критичною: у фінансових установах, на контрольно-пропускних пунктах, в системах контролю доступу та мобільних додатках.

Серед основних викликів цих технологій – забезпечення точності та швидкості обробки, а також врахування етичних питань, пов'язаних з конфіденційністю та приватністю даних.

Актуальність роботи обумовлена зростаючою популярністю розпізнавання обличчя як біометричного підходу та як одного із найефективніших і найзручніших у користуванні. Дана робота є актуальною тому, що сучасні алгоритми та попередньо натреновані моделі дозволяють з високою точністю розпізнавати обличчя, однак значним їхнім недоліком є, зокрема, вразливість до атак із використанням фальшивих облич. Однак відкритого рішення для розпізнавання живих облич не існує, тому актуальним є розроблення власного рішення, щоб задовільнити потреби безпеки.

Однією з галузей, що потребує системи для розпізнавання осіб і надання їм відповідних прав, є Internet of Things (IoT). Актуальність розпізнавання обличчя в Internet of Things (IoT) визначається кількома факторами: зростання IoT, підвищення рівня безпеки, зручність для користувачів, покращення рішень на основі даних. Отже, розпізнавання обличчя в IoT має значення як для забезпечення безпеки та зручності для користувачів, так і для покращення функціональності та ефективності систем IoT в цілому.

Система розпізнавання облич людини у відеопотоці може бути ключовим компонентом в оптимізації контролю якості потокової інформації щодо перетину держкордону з використанням методів штучного інтелекту.

Інтеграція системи розпізнавання облич дозволяє автоматизувати процес виявлення та ідентифікації осіб, що перетинають кордон, що робить контроль більш ефективним та швидким. Застосування методів штучного інтелекту дозволяє вдосконалити алгоритми розпізнавання облич та підвищити точність виявлення.

Поєднання цих технологій дозволяє створити систему, здатну в реальному часі виявляти та ідентифікувати осіб на відеопотоці, оптимізуючи тим самим контроль якості потокової інформації щодо перетину держкордону. Такий підхід дозволяє забезпечити високу ефективність та надійність контролю на державному кордоні, зменшуючи можливість пропуску небажаних осіб або предметів через кордон.

Інтеграція системи розпізнавання облич може значно підвищити ефективність інформаційних систем, що використовуються Державною прикордонною службою України (ДПСУ), дозволяючи автоматично виявляти та відстежувати осіб, які перетинають кордон або здійснюють інші дії, які потребують уваги.

Крім того, застосування системи розпізнавання облич може допомогти в реагуванні на потенційні загрози та вияв-

лення осіб, які перебувають у розшуку або мають злочинні наміри.

Спостережуваність та керованість програмних комплексів розпізнавання облич людей у відеопотоці є обов'язковою характеристикою таких систем і забезпечується на етапі обрання та реалізації архітектурних рішень щодо створення відповідних програмних систем. Такий підхід до побудови програмних комплексів забезпечує підвищений рівень безпеки та контролю на кордоні.

Мета роботи – покращення процедури ідентифікації та спрощення виявлення осіб на відео для працівників відділу безпеки підприємств шляхом впровадження методів розпізнавання із живих облич (liveness detection).

Розпізнавання облич, особливо в комбінації з виявлення живих облич, було дуже поширеною темою досліджень останній час, зокрема, як аналог розпізнавання відбитків пальців та сітківки ока. Але в розпізнаванні облич підходи до вирішення питання біометричного застосування дещо відрізняються. LD – це процес диференціації простору ознак на живі і неживі. Зловмисники намагатимуться проникнути в систему через велику кількість spoof-атак, і саме LD може помітно покращити безпеку застосування розпізнавання облич у біометричних цілях. У розпізнаванні облич типові методи атак можна розділити на кілька категорій, їхня класифікація ґрунтується на тому, у який спосіб системі верифікації надається хибна інформація. Наприклад, фотографія, зображення обличчя, записане відео, 3D-моделі обличчя з можливістю моргання і руху губ, 3D-моделі обличчя з різними виразами тощо[2].

Хоча найкращі сучасні реалізації показують неймовірні результати, досі є місце для росту, а також для покращення доступності. Якість відео, шум, інтенсивність освітлення значно впливають на якість роботи моделей. Також методи добре протидіють відомим способам атаки, але проти раніше не бачених мало ефективні. Зрештою, однією з ключових проблем, яка потребує врегулювання - наявність якісних відкритих датасетів: їх потенційне створення суперечить приватності персональних даних багатьох людей.[3]

У межах цієї роботи було обрано стратегію розробки системи розпізнавання face recognition з акцентом на виявленні живих облич (liveness detection), що є ключовим елементом для підтримання високого рівня безпеки.

Аналіз алгоритмічних та технічних рішень

Розглянемо відомі алгоритмічні рішення для виявлення обличчя.

Haarcascade (Haar Cascade Classifier) – використовує каскади Хаара для ідентифікації облич. Він базується на концепції "прямокутників", які є простими формами (на кшталт країв, ліній), що використовуються для визначення облич на зображенні. Імплементація: Навчання відбувається за допомогою методу "посиленого навчання" (AdaBoost). Система використовує багато різних форм та обирає найкращі для ефективного розпізнавання облич.[4]

HOG (Histogram of Oriented Gradients) - використання гістограми орієнтованих градієнтів для виявлення облич. Цей алгоритм розділяє зображення на невеликі області та обчислює гістограми градієнтів для кожної області, щоб створити опис обличчя. HOG часто використовується у поєднанні з класифікатором, як-то SVM (Support Vector Machine), для визначення чи присутнє обличчя у даній області.[5]

MMOD (Max-Margin Object Detection) є методом глибокого навчання (DL) для детекції об'єктів. Він використовує концепцію "максимального відступу" для визначення кращого розміщення коробки детекції навколо обличчя. MMOD зазвичай включає в себе глибокі конволюційні нейромережі для вивчення характеристик облич. Він є більш точним у порівнянні з класичними методами, але вимагає більшої обчислювальної потужності.[6]

SSD (Single Shot MultiBox Detector) є ще одним методом глибокого навчання, який дозволяє виявляти об'єкти на зображеннях за один прохід. Він ефективно визначає різні об'єкти та їхнє розташування на зображенні. SSD використовує ряд конво-

люційних шарів для виявлення об'єктів різного розміру. Цей метод швидкий та ефективний, особливо у реальному часі.[7]

Кожен із цих методів має свої переваги та недоліки, і вибір конкретного алгоритму залежить від конкретних вимог до точності, швидкості та ресурсів обчислювальної системи.

У випадку цієї розробки, SSD виявився найбільш ефетивним з точки зору співвідношення обчислювальної складності до якості ідентифікації, тому для реалізації ПЗ буде використано саме його.

Розглянемо алгоритми розпізнавання облич.

Eigenfaces використовує метод головних компонентів (PCA) для зменшення розмірності простору облич. Високовимірні дані облич перетворюються на набір вагових значень, які представляють ключові риси обличчя. Імплементація Eigenfaces включає навчання на наборі облич, щоб визначити головні компоненти. Кожне нове обличчя проєктується на цей простір для порівняння або розпізнавання.[8]

Цей метод придатний для систем із невеликою кількістю даних і меншими обчислювальними ресурсами, однак коли можлива велика кількість облич і якість розпізнавання, він мало придатний.

Fisherfaces використовує лінійний дискримінантний аналіз (LDA) для оптимізації розрізнення між класами облич. Вони зосереджені на максимізації відстані між категоріями облич, одночасно мінімізуючи варіації всередині кожного класу. Fisherfaces навчається на наборі облич, щоб визначити оптимальні лінійні комбінації, які розмежовують різні класи облич. [9]

Цей метод ефективніший за Eigenfaces для ситуацій, де класи облич мають високий рівень варіативності і загалом має більшу точність та обчислювальні витрати.

LBPH (Local Binary Patterns Histograms) використовує локальні двійкові шаблони для опису характеристик обличчя. Цей метод аналізує кожен піксель зображення, порівнюючи його з сусідніми пікселями, та кодує ці відносини у двійковому форматі.[10]

Імплементація: LBPH обчислює гістограми локальних двійкових шаблонів для різних частин обличчя, а потім ці гістограми використовуються для порівняння та розпізнавання облич. Цей метод ефективний для варіацій у освітленні та виразах облич.

Алгоритми глибокого навчання, переважно конволюційні нейромережі (CNN), також використовуються для розпізнавання облич. Вони автоматично вивчають риси облич з великих наборів даних, виявляючи складні шаблони та властивості. Вони вважаються одними з найточніших методів, але вимагають великих обчислювальних ресурсів та великої кількості даних для ефективного навчання.[11]

Кожен із цих методів має свої особливості та придатний для різних сценаріїв застосування. Наприклад, LBPH може бути кращим вибором для систем з обмеженими ресурсами, тоді як глибоке навчання є ідеальним для складних застосувань із великими наборами даних.

Для реалізації було обрано CNN алгоритм для кращої якості ідентифікації, тому для реалізації ПЗ буде використано саме його.

Виявлення живих облич (LD) є важливою частиною систем біометричної безпеки, яка допомагає визначити, чи об'єкт, що сканується, є справжньою людиною, а не фотографією, відео чи іншою підробкою. Розглянем кілька методів, що використовуються для liveness detection:[2]

Аналіз руху зводиться до розрізнення патерну руху між 3D та 2D об’єктами, які зазвичай є зображеннями, що використовуються для обману системи. Обов’язковою для методу є наявність відеопотоку, проте інтеракція з користувачем тут не потрібна, що в поєднанні з високою ефективністю розпізнавання 2D зображень є значними перевагами. До недоліків можна віднести потребу у високоякісному відео, а також погіршення ефективності, якщо в сцені немає активного руху.

Аналіз текстури використовує той факт, що надруковане чи виготовлене іншим чином, або вставлене зображення матиме дефекти, розмитість, чи можливо навіть інше оточення у порівнянні з рештою

зображення. Цей підхід легкий в імплементації і не потребує взаємодії з користувачем, він охоплює більшу множину можливих атак. Недоліки цього методу знову зводяться до потреби у високоякісних зображеннях, до того ж ефективність стрімко падає, якщо патерн відмінностей в текстурах не зустрічався в навчальних даних, а це ставить жорсткі вимоги до об'єму і різноманітності датасету.

Виявлення міміки можна поділити на два типи. До першого відносять виконання користувачем певного руху (повернути голову вправо), або певного виразу обличчя(посміхнутись). Якщо цю вказівку виконано, то верифікація успішна. Це єдиний з розглянутих методів, який вимагає активної взаємодії з користувачем, і, відповідно, недоліком є впровадження людського фактору. Друга категорія полягає у виявленні притаманної усім живим людям міміки, наприклад - кліпання очима. Ускладнення, яке приносить цей метод, - потреба дотатково виявити певні частини на обличчі, такі як очі абощо. Загалом, міміка ефекктивніша в порівнянні з іншими методами, але потребує додаткових засобів для реалізації.

Для реалізації було обрано аналіз тестур та патернів. Оскільки готових відкритих рішень немає, буде розроблено власну CNN мережу, через простоту в імплементації і хороші результати[12], які вона показує у порівнянні з аналогічними моделями.

Отож, для розроблення було обрано SSD модель для виявлення облич та CNN моделі для розпізнавання та виявлення живих облич. Це створює високі вимоги для апаратного забезпечення, щоб процес міг проходити в реальному часі.

Архітектура програмного забезпечення

Для розроблення додатка з інтерфейсом важливо відокремити представлення, логіку та дані, для створення більш гнучкої та модульної архітектури. Однак, враховуючи, що всі дані, якими оперує додаток, – це мітка особи та зображення для її ідентифікації, а одна з функціональних вимог – можливість захопити нове обличчя для ідентифікації, то ці дані краще зберігати локально, ніж в окремій базі даних. Тож, враховуючи особливості мови Python та велику кількість біліотек, будуть використані елементи патернів Adapter та Decorator(Wrapper).

Розглянемо діаграму компонентів (рис. 1).

Оскільки бібліотека Qt призначена для розробки інтерфейса методологією What You See Is What You Get, використання інших архітектурних патернів не є доречним.

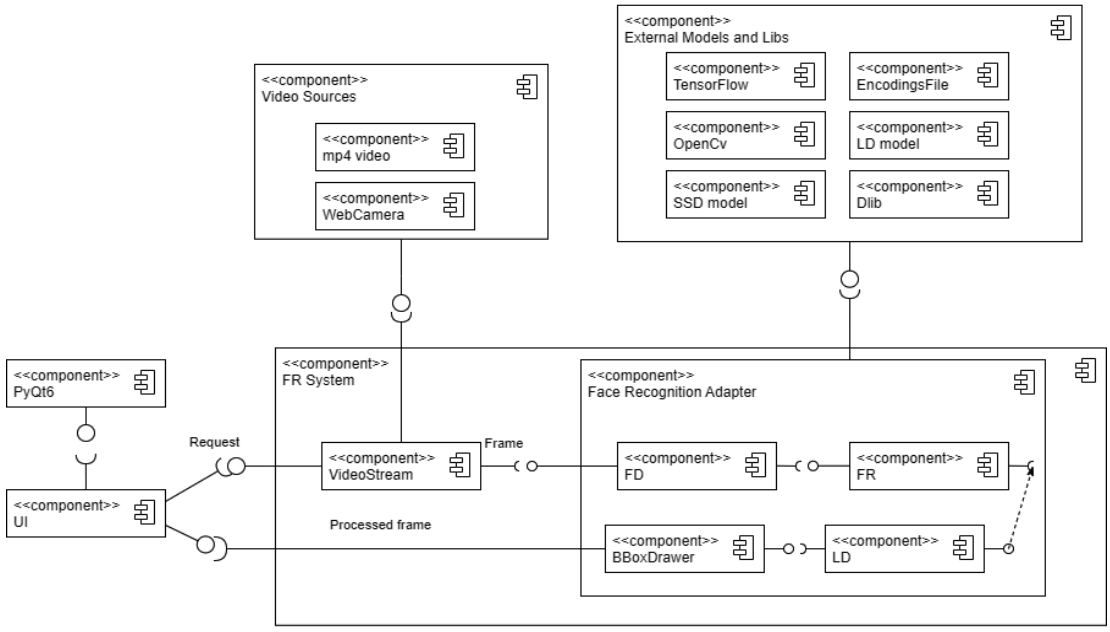

Рис. 1. Діаграма компонентів

Інтерфейс матиме вигляд декстопного дотатку, реалізованого за допомогою PyQt6 з головного екрану, вікон відеопотоку, а також діалогових вікон, які даватимуть користувачеві можливість отримувати доступ до всього функціоналу ПЗ, а також вводити необхідні дані і бачити результат роботи програми.

Бізнес-логіка буде завантажувати кодування облич, моделі для FD, FR та LD, а також отримувати відеопотік, оброблення якого відбуватиметься в окремому потоці від решти завдань. Для оброблення, отримання збереження та виведення кадрів використовуватимуться можливості OpenCV. Дані про кожне захоплене обличчя (20 зображень) будуть зберігатися в окремій теці, а для розпізнавання будуть виділятися характеристики і закодовуватись у вектор довжиною 128 засобами бібліотеки Dlib.

Опис використаних алгоритмів та архітектур нейронних мереж

Для створення відеопотоку було використано можливості OpenCV, яка дозволяє захоплювати відеопотік як з вебкамер, так і з відео. Оскільки для оброблення в реальному часі необхідний швидкий доступ до кадрів потоку з мінімальною латентністю, було використано модуль Tread. Потік постійно оновлюється паралельно виконанню обробки попередньо взятого кадру, що значно зменшує затримки.

Для знаходження облич на кадрі з відеопотоку було використано нейромережу SSD архітектури. Підхід SSD базується на згортковій мережі, яка створює колекцію BBox та оцінок наявності екземплярів об'єкту класу у цих рамках. BBox подібні до тих, що застосовуються в Faster R-CNN, тож враховують співвідношення сторін для прямокутника з об’єктом, тому враховують співвідношення сторін, однак застосовуються до карт властивостей(feature maps) у різному маштабі. Використана нейронна мережа була натренована для розпізнавання облич у Caffe framework, тому її можна завантажити за допомогою DNN модуля бібліотеки OpenCV. Ця модель поєднує високу якість із високою продуктивністю, дозволяючи розпізнавати обличчя під різними кутами. Детальну архітектуру шарів нейромережі можна побачити на рис. 2.

Для розпінавання обличчя використовуються можливості бібліотеки Dlib. За допомогою MMOD (Max-Margin Object Detection)[6] виявляє 68 точок[14] на обличчі, після чого вони закодовуються у вектор властивостей довжиною 128 [15] за допомогою ResNet моделі, що дає можливість порівнювати зображення із раніше збереженим. Знайшовши різницю кодувань, отримаємо відстань між зображеннями, і чим вона менша, тим більша ймовірність, що це одна особа. [14] Пакет face_recognition дає зручний інтерфейс для такого рішення, побудованого на основі Dlib.

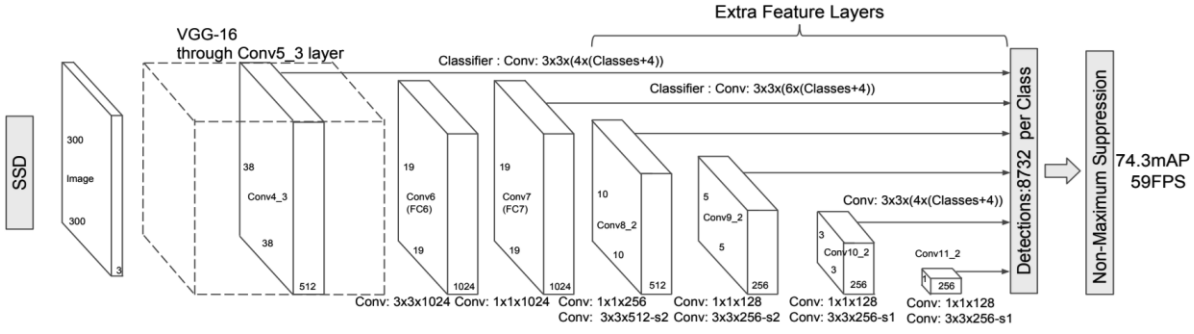

Рис. 2. Архітектура SSD моделі

Для LD було натреновано власну нейронну мережу. Враховуючи високу ефективність і швидкодію CNN моделей у цій задачі [13], було обрано саме таку архітектуру. Задача розглядалась як бінарна класифікація. Вхідне зображення для навчання мало розмір 150 на 150 пікселів. Для навчання використовувалась бібліотеки TensorFlow та Keras. Вихідний шар має оцінку йомовірності того, що зображення міс-

тить живе обличчя. Для оптимізації використовувався алгоритм Adam.

Опис датасету

Для розпізнавання облич необхідно зберегти мітку, присвоєну персоні, та її зображення. Чим більша кількість різнотипних зображень, тим краща якість розпізнавання, але тим довше процес захоплення. Тому було обрано кількість зображень у 20. Оскільки для відображення в інтерфейсі та кодування зображень та одночасної можливості захоплення в реальному часі необхідний постійний доступ до зображень, то було вирішено зберігати їх локально.

Зображення кожного захопленого обличчя зберігаються у відповідній папці та з відповідною міткою. Для їх редагування, зміни та доступу використовується бібліотека os.

Для навчання нейронної мережі для LD потрібний розмічений датасет із розмічених даних 2 типів – справжніх та несправжніх облич. Дані були отримані шляхом відеозапису обличчя (справжні) та запису відео, продемонстрованого з телефону на іншу камеру(неживі). Додатково було взято дані із датасету CelebA-Spoof[16], під час навчання відбувалась їхня аугментація. Таким чином, покрита велика множина можливих атак: від демонстрації запису відео чи окремого зображення до створення фейкових (роздруківки, фотографії, 3d моделі) облич. Графік процесу навчання видно на рис. 3.

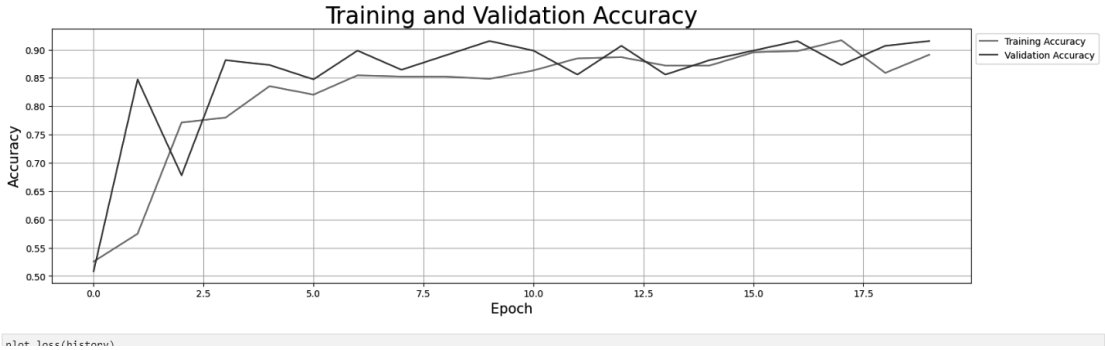

Рис. 3. Графік навчання LD моделі

Шляхи вдосконалення та подальшого розвитку

Можна виділити 3 основні напрямки подальшого розвитку:

1. Впровадження процесу інтеграції та розгортання додатку на сервері;
2. Покращення ефективності роботи додатку із можливістю роботи в реальному часі на пристроях без GPU;
3. Покращення LD.

Для першого необхідно розмістити додаток на сервері, який матиме доступ до вебкамер, а також інтегрувати взаємодію із зовнішньою базою даних.

Для другого напрямку необхідне глибше використання технологій паралельного програмування, а також можлива заміна використаних сетодів на менш якісні, однак оптимальніші з точки зору обчислювальних витрат.

Третій напрямок полягає у розробленні датасету із охопленням більшої кількості видів атак, охопленням більшої кількості підходів до LD та покращенням архітектури нейронної мережі.

Висновки

В результаті дослідження було спроєктовано декстопний застосунок, призначений для виявлення, розпізнання та виявлення живих облич у відеопотоці. Результатом розробки є покращення процедури ідентифікації або спрощення виявлення осіб на відео для працівників відділу безпеки підприємств шляхом впровадження методів розпізнавання живих облич (liveness detection), збільшення їх доступності за рахунок інтерфейсу.

За головне середовище розробки обрано Python. Як середовища розробки було обрано IDE PyCharm та Jupyter-notebook.

Для реалізації проєкту використовувалось багато бібліотек, зокрема, Keras та Dlib для нейронних мереж, OpenCV для відеопотоків та обробки зображення, а також Qt для інтерфейсу.

Було проаналізовано архітектурні рішення і обрано архітектурні патерни. Також було проаналізовано архітектури нейромереж, відомі алгоритмічні рішення і з-поміж них було обрано оптимальні для поставленої задачі.

Проаналізувавши відомі успішні програмні продукти, було виділено ніші, які потребують представлення нового рішення. На їх основі було розроблено функціональні та нефункціональні вимоги, поставлені задачі.

Потім було здійснено аналіз бізнес-процесів програмного забезпечення, реалізовано планування архітектури та тестування програмного забезпечення на відповідність вимогам.

Новизна роботи полягає в модифікації процесу розпізнавання облич у відеопотоці шляхом впровадження власної моделі Liveness Detection.

Практична значущість розробленої системи полягає в наступному:

- Система розпізнавання обличчя дозволяє автоматизувати процеси вхідного контролю та відвідування, забезпечуючи безпечний та зручний доступ для працівників.
- Використання системи розпізнавання обличчя сприяє підвищенню рівня безпеки на території підприємства, запобігає несанкціонованому доступу та допомагає вчасно реагувати на події.
- Система може використовуватися для точного фіксування часу входу та виходу працівників, що полегшує облік робочого часу та покращує контроль над робочим графіком.
- Система розпізнавання обличчя дозволяє швидко та ефективно ідентифікувати осіб, що призводить до скорочення часу на реєстрацію та перевірку.

Отримані результати розв'язують поставлені задачі і досягають зазначеної мети, що були описані у технічному завданні. Однак залишається місце для покращення, особливо щодо продуктивності та якості LD.

Література

1. Implementation of face recognition and liveness detection system using TensorFlow.js [Електронний ресурс] — https://jurnal.polinema.ac.id /index.php/jip/article/view/3977/2759
2. An overview of face liveness detection [Електронний ресурс] — https://arxiv.org/ftp/arxiv/papers/1405/1405.2227.pdf
3. Face Liveness Detection Using Artificial Intelligence Techniques: A Systematic Literature Review and Future Directions [Електронний ресурс] — https://doi.org/10.3390/ bdcc 7010037
4. Face Detection Using Haar Cascade Classifiers Based on Vertical Component Calibration [Електронний ресурс] — http://hcisj.com/data/ file/article/2022031501/12-11.pdf?ckatt empt=1
5. Comparison of Viola-Jones Haar Cascade Classifier and Histogram of Oriented Gradients (HOG) for face detection [Електронний ресурс] — https://iopscience.iop.org/article/10.1088/1757-899X/732/1/012038/meta
6. Max-Margin Object Detection [Електронний ресурс] — https://arxiv.org/abs/1502.00046
7. Eigenfaces for Recognition [Електронний ресурс] — https://direct.mit.edu/jocn/article/3/1/71/3025/Eigenfaces-for-Recognition
8. Face Recognition Using Fisherface Method [Електронний ресурс] — https://iopscience.iop.org/article/10.1088/1742-6596/1028/1/012119/meta
9. LBPH-based Enhanced Real-Time Face Recognition [Електронний ресурс] — https://thesai.org/Publications/ViewPaper?Volume=10&Issue=5&Code=IJACSA&SerialNo=35

10. When Face Recognition Meets With Deep Learning: An Evaluation of Convolutional Neural Networks for Face Recognition [Електронний ресурс] — https://www.cv-foundation.org/openaccess/content_iccv_ 2015_workshops/ w11/html/ Hu_ When_Face_Recognition_ICCV_2015 _paper.html
11. Optimizing Deep CNN Architectures for Face Liveness Detection [Електронний ресурс] — https://www.mdpi.com/1099-4300/21/4/423
12. 13 Top 11 Facial Recognition Software in 2021 [Електронний ресурс] — https://www.spiceworks.com/it-security/identity-access-management/articles/facial-recognition-software/
13. face_recognition package [Електронний ресурс] — https://face-recognition.readthedocs.io/en/latest/face_recognition.html#module-face_recognition.api
14. Dlib C++ library [Електронний ресурс] — http://dlib.net/python/index.html#dlib_pybind11.face_recognition_model_v1
15. CelebA-Spoof: Large-Scale Face Anti-Spoofing Dataset with Rich Annotations [Електронний ресурс] — https://github.com/ZhangYuanhan-AI/CelebA-Spoof

Одержано: 10.04.2024
Внутрішня рецензія отримана: 21.04.2024
Зовнішня рецензія отримана: 27.04.2024

Про авторів:

[1]*Попереshняк Світлана Володимирівна*,
Кандидат фізико-математичних наук,
доцент
http://orcid.org/0000-0002-0531-9809.

[2]*Скорик Родіон Олегович*,
бакалавр
http://orcid.org/0009-0000-9547-4038.

[3] *Купцов Дмитро Володимироович*,
аспірант
http://orcid.org/0009-0009-9958-6809.

[4] *Кравченко Роман Вікторович*,
аспірант
http://orcid.org/0009-0005-8044-4414.

Місце роботи авторів:

[1, 2] Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»,
тел. +38-098-645-54-62
E-mail: spopereshnyak@gmail.com

[3,4] Інститут програмних систем НАН України.